Mots-clés CONTACT et LIAISON_UNILATER

1 But

Affecter des conditions de contact unilatéral et de frottement en mécanique (CONTACT) ou des conditions unilatérales sur les autres degrés de liberté (LIAISON_UNILATER).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des matières

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Opérande CONTACT

Mot-clé facteur utilisable pour décrire les zones soumises à des conditions de contact unilatéral avec ou sans frottement. Ces zones (une pour chaque occurrence du mot-clé facteur) comprennent chacune deux surfaces pouvant entrer en contact qui sont décrites par la donnée des mailles qui les constituent.
Ce type de conditions aux limites n'est implanté que dans les opérateurs STAT_NON_LINE [U4.51.03] et DYNA_NON_LINE [U4.53.01].

Les ensembles de mailles potentiellement en contact sont : surfaciques et linéiques en dimension 3 (QUAD9, QUAD8, QUAD4 et TRIA7, TRIA6, TRIA3 et SEG3, SEG2), linéiques et concentrées en dimension 2 (SEG3, SEG2 et POI1). Les mailles de type POI1 doivent obligatoirement être sur la surface esclave. Elles ne sont pas utilisables avec la méthode CONTINUE.

Attention :

Pour les formulations discrètes, en dimension 3, le traitement du contact avec des mailles surfaciques quadratiques de type QUAD8 nécessite de lier les nœuds milieux des côtés aux sommets de façon à avoir des résultats corrects (de même pour les mailles QUAD9 et TRIA7 associées à la modélisation COQUE_3D). Cette opération est faite automatiquement dans le code. Néanmoins, pour les calculs 3D milieux continus avec des éléments héxaèdriques quadratiques, l'utilisation d'éléments HEXA27 (à faces QUAD9) est fortement conseillée.

Les structures étudiées peuvent subir de grands glissements l'une par rapport à l'autre. Cette formulation en géométrie réactualisée existe en trois variantes différentes :
1. Les formulations discrètes (voir [R5.03.50]) qui correspondent aux méthodes CONTRAINTE, GCP, PENALISATION et LAGRANGIEN
2. La formulation continue (voir [R5.03.52]) qui correspond à la méthode CONTINUE
3. Les formulations sur les éléments XFEM (voir [R7.02.12] et [R5.03.53] pour la version grands glissements) qui correspondent à la méthode XFEM

Avant de faire un calcul avec contact utilisant le mot-clé CONTACT, il est **indispensable** d'avoir lu les documentations de référence ainsi que la documentation [U2.04.04] de conseils aux utilisateurs qui explicitent le rôle de la plupart des mots-clés décrits ci-dessous et donnent les précautions d'utilisation.
À noter que le solveur GCPC ne doit pas être utilisé avec le contact.
Toutes les méthodes de résolution du contact/frottement (formulations discrètes [R5.03.50] ou formulation continue [R5.03.52]), dites formulations « maillées » (par opposition à la formulation XFEM) reposent sur une stratégie en deux temps :
- une opération d'appariement qui consiste à trouver quelles mailles et quels nœuds sont potentiellement en situation de contact;
- une opération de résolution proprement dite qui consiste à résoudre le problème de contact unilatéral avec ou sans frottement;

2.1 Syntaxe

```
CONTACT    =   _F(
                   ◊   METHODE       =   /'CONTRAINTE'              [DEFAUT]
                                         /'GCP'
                                         /'PENALISATION'
                                         /'LAGRANGIEN'
                                         /'VERIF'
                                         /'CONTINUE'
                                         /'XFEM'

#                  si METHODE == 'PENALISATION'/'LAGRANGIEN'/'CONTINUE'/'XFEM'
                      ◊   FROTTEMENT    =   /'SANS'             [DEFAUT]
                                            /'COULOMB'
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
#               fin si


#            Si méthodes maillées (c'est à dire si METHODE != 'XFEM')
                ♦  /  MAILLE_MAIT   =  lma1,                  [l_maille]
                   /  GROUP_MA_MAIT=  lgma1,                  [l_gr_maille]
                ♦  /  MAILLE_ESCL   =  lma2,                  [l_maille]
                   /  GROUP_MA_ESCL=  lgma2,                  [l_gr_maille]

                ◊  APPARIEMENT  =  /'MAIT_ESCL'               [DEFAUT]
                                   /'NODAL'

                 ◊ NORMALE      =  /'MAIT'                    [DEFAUT]
                                   /'ESCL'
                                   /'MAIT_ESCL'

                ◊  LISSAGE      =  /'NON'                     [DEFAUT]
                                   /'OUI'

                ◊  VECT_MAIT    =  /'AUTO'                    [DEFAUT]
                                   /'FIXE'
                                   /'VECT_Y'
#               si VECT_MAIT == 'FIXE'
                   ♦  MAIT_FIXE    =  (Yx,Yy,Yz)              [R]
#               fin si
#               si VECT_MAIT == 'VECT_Y'
                   ♦  MAIT_VECT_Y  =  (Yx,Yy,Yz)              [R]
#               fin si

                ◊  VECT_ESCL    =  /'AUTO'                    [DEFAUT]
                                   /'FIXE'
                                   /'VECT_Y'
#               si VECT_ESCL == 'FIXE'
                   ♦  ESCL_FIXE    =  (Yx,Yy,Yz)              [R]
#               fin si
#               si VECT_ESCL == 'VECT_Y'
                   ♦  ESCL_VECT_Y  =  (Yx,Yy,Yz)              [R]
#               fin si

                ◊  TYPE_APPA    =  /'PROCHE'                  [DEFAUT]
                                   /'FIXE'
#               si TYPE_APPA == 'FIXE'
                   ♦  DIRE_APPA    =  (Yx,Yy,Yz)              [R]
#               fin si

                ◊  TOLE_APPA    =  -1.0                       [DEFAUT]
                                   tole                       [R]

                ◊  DIST_MAIT    =  dist_mait                  [FONCTION]
                ◊  DIST_ESCL    =  dist_escl                  [FONCTION]

#               si méthodes discrètes (c'est à dire si METHODE != 'CONTINUE')
                   ◊  DIST_POUTRE  =  /'NON'                  [DEFAUT]
                                      /'OUI'
                   ◊  DIST_COQUE   =  /'NON'                  [DEFAUT]
                                      /'OUI'
#                  si DIST_POUTRE == 'OUI' ou DIST_COQUE == 'OUI'
                      ♦  CARA_ELEM =  cara                    [CARA_ELEM]
#                  fin si
#               fin si
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                  ◊   SANS_NO       =   lno1,                  [l_noeud]
                  ◊   SANS_GROUP_NO=   lno2,                   [l_gr_noeud]

#             fin si

              ◊   TOLE_PROJ_EXT=   0.5                         [DEFAUT]
                                   tole                        [R]


#             si METHODE == 'VERIF'
                  ◊   GROUP_MA_FOND=   lgma1,                  [l_gr_maille]
                   ◊   STOP_INTERP  =   /'NON',                [DEFAUT]
                                         / 'OUI',
                   ◊   TOLE_INTERP  =   /0.,                   [DEFAUT]
                                        /tole,                 [R]
#             fin si


#             si méthodes discrètes
                  ◊   REAC_GEOM     =   /'AUTOMATIQUE'         [DEFAUT]
                                        /'SANS'
                                        /'CONTROLE'
#                 si REAC_GEOM == 'CONTROLE'
                      ♦   NB_REAC_GEOM =   /val                [I]
#                 fin si
#             fin si


#             si METHODE == 'CONTRAINTE'
                  ◊   NB_RESOL  =   /10                        [DEFAUT]
                                    /nbresol                   [I]
                  ◊   STOP_SINGULIER   =   /'OUI'              [DEFAUT]
                                           /'NON'
                  ◊   GLISSIERE     =   /'NON'                 [DEFAUT]
                                        /'OUI'
#                 si GLISSIERE == 'OUI'
                      ◊   ALARME_JEU   =   /0.,                [DEFAUT]
                                           /alarme,            [R]
#                 fin si
#             fin si

#             si METHODE == 'LAGRANGIEN'
                  ◊   NB_RESOL  =   /10                        [DEFAUT]
                                    /nbresol                   [I]
                  ◊   STOP_SINGULIER   =   /'OUI'              [DEFAUT]
                                           /'NON'
                  ◊   ITER_MULT_MAXI   =   /4                  [DEFAUT]
                                           /mult_maxi          [I]
                  si FROTTEMENT == 'COULOMB'
                      ♦   COULOMB    =   /val                  [R]
                      ◊   COEF_MATR_FROT   =   /0              [DEFAUT]
                                               /val            [R]
#                 fin si
#             fin si


#             si METHODE == 'PENALISATION'
                  ◊   E_N=   /val                              [R]
#                 si E_N non renseigné
                      ◊   NB_RESOL   =   /10                   [DEFAUT]
                                         /nbresol              [I]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                  ◊   STOP_SINGULIER   =   /'OUI'             [DEFAUT]
                                           /'NON'
                  ◊   ITER_MULT_MAXI   =   /4                 [DEFAUT]
                                           /mult_maxi         [I]
#             fin si


#             si FROTTEMENT == 'COULOMB'
                  ♦   COULOMB   =   /val                      [R]
                  ◊   COEF_MATR_FROT   =   /0                 [DEFAUT]
                                           /val               [R]
                  ♦   E_T   =   /val                          [R]
#             fin si
#          fin si


#          si METHODE == 'GCP'
               ♦   RESI_ABSO =   /val                         [R]
               ◊   REAC_ITER =   /3                           [DEFAUT]
                                 /val                         [I]
               ◊   ITER_GCP_MAXI=   /0                        [DEFAUT]
                                    /val                      [I]
               ◊   PRE_COND  =   /'SANS'                      [DEFAUT]
                                 /'DIRICHLET'
               ◊   COEF_RESI =   /1                           [DEFAUT]
                                 /val                         [R]
               ◊   ITER_PRE_MAXI=   /0                        [DEFAUT]
                                    /val                      [R]
               ◊   RECH_LINEAIRE=   /'ADMISSIBLE'             [DEFAUT]
                                    /'NON_ADMISSIBLE'
#          fin si


#          si METHODE == 'CONTINUE'
               ◊   ITER_CONT_MAXI   =   /30                   [DEFAUT]
                                        /iter_cont_maxi       [I]
               ◊   ITER_GEOM_MAXI   =   /2                    [DEFAUT]
                                        /iter_geom_maxi       [I]
               ◊   INTEGRATION      =   /'NOEUD'              [DEFAUT]
                                        /'GAUSS'
                                        /'SIMPSON'
                                        /'SIMPSON1'
                                        /'SIMPSON2'
                                        /'NCOTES'
                                        /'NCOTES1'
                                        /'NCOTES2'
               ◊   CONTACT_INIT =   /'NON'                    [DEFAUT]
                                    /'OUI'
               ◊   GLISSIERE    =   /'NON'                    [DEFAUT]
                                    /'OUI'

               ◊   RACCORD_LINE_QUAD   =   /'NON'                [DEFAUT]
                                        /'OUI'
#             si RACCORD_LINE_QUAD == 'OUI'
                  ♦   /   NOEUD_RACC    =   lno1,             [l_noeud]
                      /   GROUP_NO_RACC=    lno2,             [l_gr_noeud]
#             fin si
               ◊   FOND_FISSURE =   /'NON'                    [DEFAUT]
                                    /'OUI'
#             si FOND_FISSURE == 'OUI'
                  ♦   /   NOEUD_FOND    =   lno1,             [l_noeud]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                      /   GROUP_NO_FOND=  lno2,              [l_gr_noeud]
                      /   MAILLE_FOND  =  lma1,              [l_maille]
                      /   GROUP_MA_FOND=  lgma1,             [l_gr_maille]
#               fin si

                ◊  EXCLUSION_PIV_NUL   =  /'NON'             [DEFAUT]
                                          /'OUI'
                ◊  COMPLIANCE    =  /'NON'                   [DEFAUT]
                                    /'OUI'
                si COMPLIANCE == 'OUI'
                   ♦  ASPERITE  =  /asperite                 [R]
                   ♦  E_N       =  /e_n                      [R]
                   ◊  E_V       =  /0.                       [DEFAUT]
                                   /e_v                      [R]
#               fin si
                ◊  FORMULATION  =  /'DEPL'                   [DEFAUT]
                                   /'VITE'

                ◊  ALGO_CONT =  /'LAGRANGIEN'                [DEFAUT]
                                /'AUGMENTE'
                                /'STABILISE'
#               si ALGO_CONT == 'LAGRANGIEN'
                   ◊  COEF_REGU_CONT   =  /100.              [DEFAUT]
                                          /coef_regu_cont    [R]
#               fin si
                si ALGO_CONT == 'STABILISE'
                   ◊  COEF_REGU_CONT   =  /100.              [DEFAUT]
                                          /coef_regu_cont    [R]
                   ◊  COEF_STAB_CONT   =  /100.              [DEFAUT]
                                          /coef_stab_cont    [R]
#               fin si
                si ALGO_CONT == 'AUGMENTE'
                   ◊  COEF_REGU_CONT   =  /100.              [DEFAUT]
                                          /coef_regu_cont    [R]
                   ◊  COEF_STAB_CONT   =  /100.              [DEFAUT]
                                          /coef_stab_cont    [R]
                   ◊  COEF_PENA_CONT   =  /100.              [DEFAUT]
                                          /coef_pena_cont    [R]
#               fin si

#               si FROTTEMENT == 'COULOMB'
                   ◊  ITER_FROT_MAXI   =  /2                 [DEFAUT]
                                          /iter_frot_maxi    [I]
                   ♦  COULOMB   =  coef_coulomb              [R]
                   ◊  SANS_NOEUD_FR    =  lno1,              [l_noeud]
                   ◊  SANS_GROUP_NO_FR =  lno2,              [l_gr_noeud]
#                  si SANS_NOEUD_FR ou SANS_GROUP_NO_FR
                      ◊  EXCL_FROT_1  =(Vx,Vy,Vz)            [R]
                      ◊  EXCL_FROT_2  =(Vx,Vy,Vz)            [R]
#                  fin si
                   ◊  SEUIL_INIT   =  /0.                    [DEFAUT]
                                      /seuil_init            [R]

                   ◊  USURE  =  /'SANS'                      [DEFAUT]
                                /'ARCHARD'
#                  si USURE == 'ARCHARD'
                       ♦  K =  /k                            [R]
                      ♦  H  =  /h                            [R]
#                  fin si

                   ◊  ALGO_FROT =  /'LAGRANGIEN'             [DEFAUT]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                  /'AUGMENTE'
                                  /'STABILISE'
#                   si ALGO_FROT == 'LAGRANGIEN'
                       ◊  COEF_REGU_FROT   =  /100.            [DEFAUT]
                                              /coef_regu_frot  [R]
#                   fin si
#                   si ALGO_FROT == 'STABILISE'
                       ◊  COEF_REGU_FROT   =  /100.            [DEFAUT]
                                              /coef_regu_frot  [R]
                       ◊  COEF_STAB_FROT   =  /100.            [DEFAUT]
                                              /coef_stab_frot  [R]
#                   fin si
#                   si ALGO_FROT == 'AUGMENTE'
                       ◊  COEF_REGU_FROT   =  /100.            [DEFAUT]
                                              /coef_regu_frot  [R]
                       ◊  COEF_STAB_FROT   =  /100.            [DEFAUT]
                                              /coef_stab_frot  [R]
                       ◊  COEF_PENA_FROT   =  /100.            [DEFAUT]
                                              /coef_pena_frot  [R]
#                   fin si
#                fin si
#             fin si


#             si METHODE == 'XFEM'
                 ♦  FISS_MAIT     =  fiss_mait,                [FISS_XFEM]
                 ◊  ITER_CONT_MAXI  =  /30                     [DEFAUT]
                                       /iter_cont_maxi         [I]
                 ◊  ITER_GEOM_MAXI  =  /2                      [DEFAUT]
                                       /iter_geom_maxi         [I]
                 ◊  INTEGRATION     =  /'FPG4'                 [DEFAUT]
                                       /'GAUSS'
                                       /'FPG2'
                                       /'FPG3'
                                       /'FPG6'
                                       /'FPG7'
                                       /'NOEUD'
                                       /'SIMPSON'
                                       /'SIMPSON1'
                                       /'NCOTES'
                                       /'NCOTES1'
                                       /'NCOTES2'
                 ◊  CONTACT_INIT =  /'NON'                     [DEFAUT]
                                    /'OUI'
                 ◊  GLISSIERE    =  /'NON'                     [DEFAUT]
                                    /'OUI'
                 ◊  ALGO_LAGR    =  /'VERSION1'                [DEFAUT]
                                    /'VERSION2'
                                    /'NON'
                 ◊  COEF_REGU_CONT  =  /100.                   [DEFAUT]
                                       /coef_regu_cont         [R]
                 ◊  COEF_ECHELLE =  /1.E6                      [DEFAUT]
                                    /coef_ech                  [R]

#                si FROTTEMENT == 'COULOMB'
                    ◊  ITER_FROT_MAXI   =  /2                  [DEFAUT]
                                           /iter_frot_maxi     [I]
                    ♦  COULOMB    =  coef_coulomb              [R]
                    ◊  COEF_REGU_FROT   =  /100.               [DEFAUT]
                                           /coef_regu_frot     [R]
                    ◊  SEUIL_INIT    =  /0.                    [DEFAUT]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
                                  /seuil_init            [R]
#             fin si
#          fin si
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.2 Opération d'appariement

2.2.1 Opérandes MAILLE_MAIT/GROUP_MA_MAIT/MAILLE_ESCL/GROUP_MA_ESCL

Pour les formulations maillées (toutes hors XFEM) l'utilisateur fournit la liste des mailles de contact potentielles de la surface maître (MAILLE_MAIT ou GROUP_MA_MAIT) et de la surface esclave (MAILLE_ESCL ou GROUP_MA_ESCL). Ces mailles doivent être surfaciques ou linéiques en dimension 3 (QUAD9, QUAD8, QUAD4 et TRIA7, TRIA6, TRIA3 et SEG3, SEG2), linéiques et concentrées en dimension 2 (SEG3, SEG2 et POI1). Le nombre de mailles et de nœuds des deux surfaces peut être différent.

Attention :

*Il est important de vérifier que la connectivité de ces mailles est telle que **la normale est sortante à la structure** (pour ce faire, voir MODI_MAILLAGE mot clé ORIE_PEAU_2D, ORIE_PEAU_3D, ORIE_NORM_COQU[U4.23.04]). Par ailleurs, il faut s'assurer que les structures «ne tiennent pas que par le contact» (notamment dans le cas d'un chargement en force imposée) : les mouvements de corps rigide doivent être bloqués par des conditions aux limites appropriées. Une bonne façon de le vérifier est d'effectuer un calcul avec l'opérateur MECA_STATIQUE ou avec les opérateurs STAT_NON_LINE/DYNA_NON_LINE en enlevant les conditions de contact du chargement.*

Dans toute la suite, on utilisera le concept maître-esclave : les nœuds de la surface esclave ne peuvent pas « pénétrer » dans les facettes (ou les nœuds) de la surface maître. Dans le cas de l'appariement de type 'MAIT_ESCL', la surface maître est celle définie par 'MAILLE_MAIT' ou 'GROUP_MA_MAIT'. Dans le cas de l'appariement de type 'NODAL' (disponible uniquement pour les formulations discrètes), la surface maître est celle qui doit comporter le plus de nœuds. Si ce n'est pas le cas l'utilisateur est arrêté par un message d'erreur et invité à intervertir les deux surfaces.

Remarque :

Il est impossible de mélanger les modélisations purement bi-dimensionnelles (contraintes planes C_PLAN, déformations planes D_PLAN et axisymétriques AXIS) avec les modélisations tri-dimensionnelles. Les surfaces maître et esclave doivent être de même nature (2D/2D ou 3D/3D). Un message d'erreur vous arrêtera dans le cas contraire.
Notons qu'une poutre, une plaque ou une coque sont de dimension 3 et qu'il est donc possible de faire du contact poutre/3D ou poutre/plaque.

2.2.2 Opérande FISS_MAIT

Pour les formulations non maillées (le cas XFEM), l'utilisateur fournit la liste des fissures de contact potentielles (FISS_MAIT). Ces fissures sont issues de l'opérateur DEFI_FISS_XFEM .

2.2.3 Opérandes SANS_NOEUD/SANS_GROUP_NO

Ces opérandes permettent d'exclure des nœuds de la liste des nœuds esclaves, opération qui est recommandée pour des nœuds soumis à des conditions aux limites dans la direction attendue du contact (exemple encastrement).

2.2.4 Opérandes RACCORD_LINE_QUAD/GROUP_NO_RACC/NOEUD_RACC

Dans le cas d'une formulation continue et lorsque l'on a au sein du même modèle des maillages non compatibles avec des ordres d'interpolation EF différents (ici l'un est linéaire, l'autre est quadratique), des conditions de raccord y sont alors appliquées afin d'assurer la continuité des déplacements et éviter les « trous » en surface, ce qui peut engendrer des relations surabondantes (systèmes matriciels singuliers et donc pivots nuls).
On peut alors utiliser l'option RACCORD_LINE_QUAD='OUI' pour traiter ce conflit entre ces conditions de raccordement surfacique cinématique linéaire/quadratique et les conditions de contact.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

On renseigne alors la liste des nœuds sur lesquels portera le traitement théorique précisé dans la documentation de référence [R5.03.52] par l'intermédiaire des mot-clés `GROUP_NO_QUAD` ou `NOEUD_QUAD`.

2.2.5 Opérandes FOND_FISSURE/GROUP_NO_FOND/NOEUD_FOND/GROUP_MA_FOND/ MAILLE_FOND

Dans le cas d'une formulation continue et lorsqu'un modèle comporte un fond de fissure avec l'utilisation de la technique de Barsoum pour le calcul des coefficients de concentration des contraintes, on peut avoir des problèmes d'incompatibilités lors de la résolution du problème tangent numérisé du contact entre les lèvres de fissure. En effet :

- l'interpolation polynomiale n'étant plus quadratique à cause du déplacement des nœuds milieux au quart, on doit veiller à contrôler l'influence des éléments de Barsoum sur l'intégration des termes de contact. Les schémas d'intégration de type Simpson ou Newton-Cotes avec subdivision en sous-éléments d'intégration virtuels permettent d'intégrer proprement les contributions élémentaires des éléments de contact en fond de fissure.
- les degrés de liberté de contact aux nœuds de fond de fissure n'ont aucune signification physique car appartenant à la fois aux surfaces maître et esclave.
- la transformation géométrique en certains nœuds particuliers des mailles modifiées en fond de fissure est singulière.

On peut alors utiliser l'option `FOND_FISSURE='OUI'` pour traiter ces problèmes d'incompatibilités entre les conditions de contact et la technique de Barsoum en fond de fissure.
On renseigne la liste des nœuds (2D/3D) ou des mailles (3D) sur lesquels portera le traitement théorique précisé dans la documentation de référence R5.03.52 par l'intermédiaire des mot-clés `GROUP_NO_FOND`, `NOEUD_FOND`, `GROUP_MA_FOND` ou `MAILLE_FOND`.

2.2.6 Opérandes SANS_NOEUD_FR/SANS_GROUP_NO_FR

Pour la formulation continue, ces mots-clefs permettent à l'utilisateur d'exclure des directions de frottement qui risquent d'entrer en conflit avec d'autres conditions aux limites de Dirichlet. Les directions de frottement exclues sont indiqués par `EXCL_FROT_1` (en 2D) et `EXCL_FROT_1/EXCL_FROT_2` (en 3D). L'exclusion de la direction de frottement permet néanmoins de garder le caractère contactant d'un nœud.

2.2.7 Opérande EXCLUSION_PIV_NUL

Pour la formulation continue, ce mot-clef activé (`EXCLUSION_PIV_NUL='OUI'`) permet d'exclure automatiquement d'éventuelles redondances entre les conditions aux limites de Dirichlet et les conditions de contact. Ce mécanisme ne permet pas de détecter tous les cas, en particulier, il est inefficace sur les conditions aux limites non homogènes, impliquant plus de deux nœuds ou non écrites dans le repère principal (X, Y, Z). En pratique, il n'est donc efficace que pour `DDL_IMPO` et `LIAISON_DDL`. Ce traitement peut s'avérer coûteux lorsque le nombre de nœuds esclaves est grand.
L'utilisateur peut utiliser la fonctionnalité `SANS_NOEUD` / `SANS_GROUP_NO` pour indiquer lui-même les nœuds susceptibles d'être en conflit.

2.2.8 Opérande APPARIEMENT

Dans le cas des formulations discrètes l'appariement peut être nœud-facette (`'MAIT_ESCL'`) ou nodal (`'NODAL'`). Pour l'appariement nodal on écrit une relation de non pénétration entre un nœud maître et un nœud esclave, alors que pour l'appariement nœud-facette on écrit cette relation entre un nœud esclave et sa projection sur la maille maître la plus proche (voir [R5.03.50] pour les détails de la méthode d'appariement).

L'appariement nodal est déconseillé car la méthode nœud-facette est plus générale et est la seule à permettre de prendre en compte les grands glissements de façon précise.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

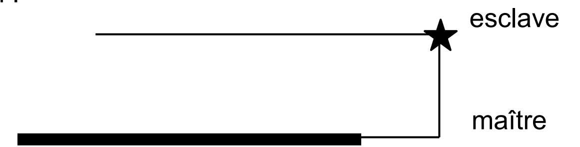
2.2.9 Opérandes TYPE_APPA/DIRE_APPA

Le choix de la maille maitre appariée au nœud esclave se fait par une opération de minimisation de la distance entre le nœud esclave et les mailles maitres. C'est l'option par défaut `TYPE_APPA='PROCHE'`. Cette procédure peut provoquer des oscillations, par exemple, si la projection sur une maille n'est pas unique, ce qui peut être le cas si la maille maître est courbe.
Pour éviter ces oscillations, l'utilisateur peut renseigner une direction d'appariement fixe via l'option `TYPE_APPA='FIXE'`, la direction étant alors donnée par un vecteur dans `DIRE_APPA`.

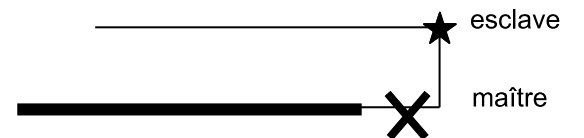
2.2.10 Opérandes TOLE_APPA et TOLE_PROJ_EXT

Lors de la recherche de la maille maitre appariée au nœud esclave courant, il est possible de restreindre le choix par utilisation du mot-clef `TOLE_APPA`. Si `TOLE_APPA=-1` (valeur par défaut), alors toutes les mailles maitres données dans la zone de contact sont susceptibles d'être appariées avec le nœud esclave. Si `TOLE_APPA=val` avec `val` un réel positif, alors seules les mailles maitres situées en 3D dans la sphère, en 2D dans le cercle de rayon `val` centrée au nœud esclave peuvent être appariés.

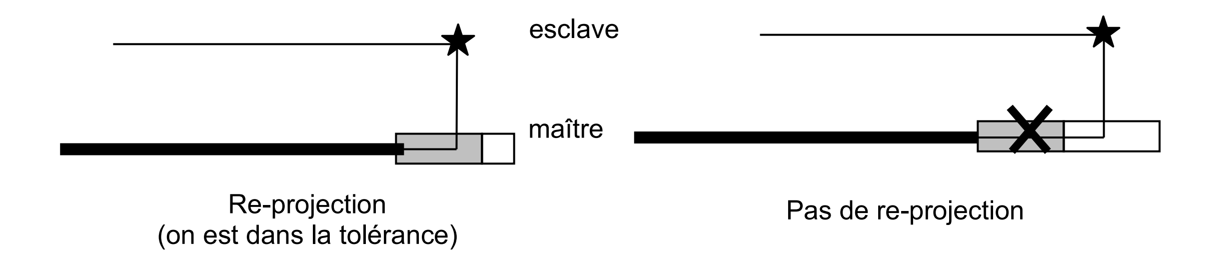
Dans certaines conditions, *Code_Aster* détecte du contact entre deux surfaces alors qu'il y en a pas. Le problème vient d'abord d'une définition incorrecte et imparfaite des surfaces susceptibles d'entrer en contact. Prenons le cas du contact en 2D (les surfaces de contact sont donc des segments).
Code_Aster procède à une re-projection sur la surface maître lorsqu'un nœud esclave se projette en dehors de celle-ci :



Une solution consiste à interdire cette re-projection :



Cependant, cette solution ne tient pas compte des cas limites et peut provoquer des interpénétrations intempestives si jamais le maillage n'est pas « optimal » (ce qui est difficile à assurer dans le cadre des grandes transformations). On a donc opté pour une solution intermédiaire en limitant l'extension de la surface maître lors de la reprojection.



La valeur limite de cette re-projection est fixée par le mot-clef `'TOLE_PROJ_EXT'` *qui prend pour argument la valeur (rapportée à l'élément de référence) de l'extension de la maille maître dans laquelle on autorise la re-projection. Par défaut, cette valeur est fixée à 0,50. Ce qui signifie que tout nœud esclave se projetant à plus de 25% à droite ou à gauche (dans le cas d'un segment dont l'élément de référence est de longueur 2, cf. [R3.01.01]) de la longueur de la maille maître ne sera pas reprojeté. Pour interdire complètement la re-projection, il suffit de fixer* `TOLE_PROJ_EXT` *négatif. Cet opérateur est valable en 2D et en 3D (dans ce dernier cas, il s'agit de l'extension d'une maille surfacique de contact).*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.2.11 Choix des normales

Par défaut, la normale prise par *Code_Aster* pour évaluer le jeu entre les deux surfaces est la normale extérieure à la maille maitre appariée au nœud esclave. Il est toutefois possible de changer ce choix.

2.2.11.1 Type de normale (NORMALE)

En premier lieu, il est possible de choisir le type de la normale:

- la normale extérieure à la maille maître (NORMALE='MAIT', par défaut)
- la normale intérieure à la maille esclave (NORMALE='ESCL')
- une moyenne entre les deux normales maitre et esclave (NORMALE='MAIT_ESCL')

2.2.11.2 Choix des normales maitre ou esclave (VECT_MAIT/VECT_ESCL)

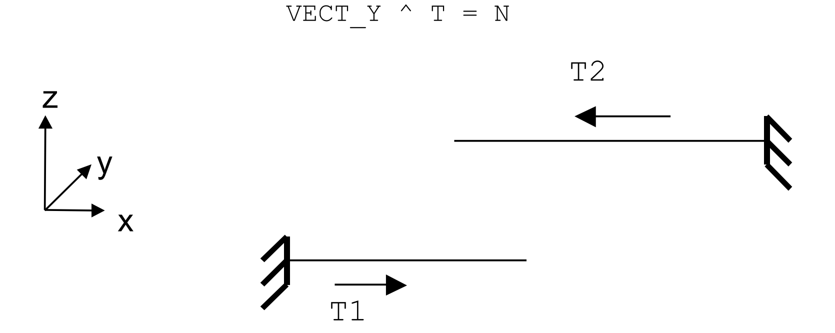
Le choix de la normale (sur la maille esclave ou sur la maille maitre) peut se faire de plusieurs manières différentes :

- automatiquement, la normale est calculée par l'utilisation des fonctions de forme de l'élément, c'est l'option 'AUTO'.
- fixe et donnée directement par l'utilisateur, c'est l'option FIXE. On entre alors la normale avec le mot-clef MAIT_FIXE ou ESCL_FIXE.
- variable et donnée indirectement par la tangente, c'est l'option VECT_Y. L'utilisateur donne alors la direction du second vecteur tangent par le mot-clef MAIT_VECT_Y ou ESCL_VECT_Y. Le code reconstitue alors la normale à partir de la première tangente de la maille.

Une direction fixe (option VECT_ESCL='FIXE') est nécessaire si le nœud esclave est de type POI1 et qu'on utilise l'option NORMALE='ESCL' ou NORMALE='ESCL_MAIT'.

Une direction variable donnée par la seconde tangente (VECT_MAIT='VECT_Y') est particulièrement utilisée dans le cas des poutres qui ne se déforme que dans un plan. Dans ce cas, l'utilisation de l'option de type VECT_Y permet de réactualiser continument la normale pendant que la poutre se déforme.

VECT_Y ^ T = N



Comme ici T1=(1,0,0) et T2=(-1,0,0), avec VECT_Y =(0,0,1), on obtient la normale souhaitée pour chaque poutre : N1=(0,1,0) et T2=(0,-1,0). Attention au fait que tout ceci est entièrement lié à l'orientation de chaque poutre.
On peut fixer l'orientation de chaque poutre à l'aide du mot-clé ORIE_LIGNE de l'opérateur MODI_MAILLAGE [U4.23.04]

2.2.11.3 Lissage des normales (LISSAGE)

L'opérande LISSAGE permet de lisser les normales aux surfaces de contact intervenant dans le calcul de la matrice de contact. On notera Q un nœud quelconque des surfaces de contact (maître ou esclave), P un nœud de la surface esclave et M le nœud maître obtenu par projection du nœud P.

Le lissage se fait en deux étapes :
- *la première étape du lissage consiste à effectuer une moyenne des normales aux mailles qui contiennent le nœud Q,*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- *la seconde étape consiste à calculer une moyenne des normales aux sommets de la maille contenant M. Cette moyenne étant pondérée par les fonctions de forme associées à M.*

Le lissage prend en compte les options de normales décidées par les mots-clés `VECT_MAIT` et `VECT_ESCL`.

2.2.12 Modification du jeu

2.2.12.1 Opérandes DIST_MAIT/DIST_ESCL

Ces opérandes permettent de prendre en compte des « trous » ou des « bosses » non maillés dans le cas des formulations maillées (hors `XFEM`) ou l'épaisseur des coques (les relations de contact sont écrites entre les deux feuillets moyens) pour les surfaces maîtres (`DIST_MAIT`) ou esclaves (`DIST_ESCL`). On compte la distance positivement dans le sens de la normale sortante à la structure (cf. [R5.03.50]). Les grandeurs renseignées sont nécessairement des fonctions des variables d'espace. Si l'utilisateur désire une valeur fixe, il doit définir une fonction constante (voir l'opérateur `DEFI_CONSTANTE`).

2.2.12.2 Opérandes DIST_POUTRE/DIST_COQUE/CARA_ELEM

Semblables aux deux mots-clés précédents, les mot-clefs `DIST_POUTRE/DIST_COQUE` (et le renseignement obligatoire du mot_clef `CARA_ELEM` qui leur est associé), permet d'introduire un jeu fictif qui repose sur la description de l'élément de structure dans le `AFFE_CARA_ELEM` que l'utilisateur a utilisé :

- pour les éléments de poutre (modélisation `POU_*)`, le mot-clef `DIST_POUTRE` autorise le code à prendre un jeu supplémentaire correspondant au rayon de la section **circulaire** de la poutre
- pour les éléments de plaque ou de coque (modélisation `DKT` ou `COQUE_3D` par exemple), le mot-clef `DIST_COQUE` autorise le code à prendre un jeu supplémentaire correspondant à la demi-épaisseur autour du feuillet moyen d'une coque

2.3 Choix des algorithmes et de la formulation

Des opérandes permettent de sélectionner une méthode de calcul suivant le type de contact (2D/3D et avec ou sans frottement) que l'on veut traiter.

2.3.1 Choix de la formulation pour le contact

Pour sélectionner le type de formulation, on utilise le mot-clef `METHODE`.

```
◊   METHODE   =   /'CONTRAINTE'              [DEFAUT]
                  /'GCP'
                  /'PENALISATION'
                  /'LAGRANGIEN'
                  /'VERIF'
                  /'CONTINUE'
                  /'XFEM'
```

Cet opérande permet d'utiliser les différentes méthodes de résolution:

- `CONTRAINTE` : par défaut on traite le problème du contact unilatéral exact sans frottement avec la méthode des contraintes actives (voir [R5.03.50])
- `GCP` : c'est une méthode itérative proche de la méthode `'CONTRAINTE'` mais qui est particulièrement adaptée aux cas où le nombre de liaisons de contact est très élevé.
- `PENALISATION` : la méthode pénalisée permet de traiter soit
  - des problèmes de contact pénalisé sans frottement 2D ou 3D si on renseigne `E_N`
  - des problèmes de contact avec frottement en 2D ou 3D avec une pénalisation sur les termes de frottement uniquement si on renseigne `E_T` et une pénalisation sur les termes de contact et de frottement si on renseigne `E_T` et `E_N.` (voir [R5.03.50])

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- LAGRANGIEN : la méthode lagrangienne permet de traiter de façon exacte par multiplicateurs de Lagrange des problèmes de contact avec ou sans frottement en 2D et 3D (voir [R5.03.50])
- VERIF : la méthode de vérification permet de contrôler si deux surfaces s'interpénètrent ou pas sans imposer les conditions de contact. C'est donc une méthode qui ne se préoccupe que de l'aspect géométrique et qui est peu coûteuse en termes de temps CPU. On peut l'utiliser par exemple pour contrôler que les deux lèvres d'une fissure ne s'interpénètrent pas.
- CONTINUE : la méthode continue permet de traiter de façon exacte, par multiplicateurs de Lagrange augmentés des problèmes de contact avec ou sans frottement en 2D et 3D
- XFEM : la méthode XFEM est une méthode continue permettant d'activer le contact/frottement sur les lèvres de la fissure

2.3.2 Méthode VERIF

Cette méthode réalise un contrôle de l'interpénétration de deux surfaces sans imposer les conditions de contact (s'il y a interpénétration, elle restera). S'il y a interpénétration, on aura une alarme. Le paramètre STOP_INTERP permet d'arrêter le calcul au lieu d'alarmer l'utilisateur. TOLE_INTERP règle la valeur d'interpénétration (homogène à une longueur).

```
◊   STOP_INTERP  =  /'NON',           [DEFAUT]
                    /'OUI',
◊   TOLE_INTERP  =  /0.,              [DEFAUT]
                    /tole,            [R]
```

On peut indiquer le fond de fissure par le mot-clef GROUP_MA_FOND. Ce mot-clef permet d'exclure du contact tous les nœuds qui se trouvent en fond de fissure (et qui risqueraient donc d'être communs à la surface maitre et la surface esclave).

2.3.3 Méthode CONTRAINTE

Cette permet de résoudre des problèmes de contact sans frottement de manière exacte (aucune interpénétration n'est tolérée). Elle est particulièrement rapide et robuste, sa convergence est prouvée. La résolution du système d'inéquations résultant du contact s'appuyant sur la construction explicite d'un complément de Schur, son utilisation est limitée à quelques centaines de liaisons de contact. Au-delà les coûts mémoire et calcul deviennent trop important.

2.3.4 Méthode GCP

Cette méthode permet de résoudre des problèmes de contact sans frottement. Elle résout avec une précision réglable (éventuellement très élevée) les conditions de contact à l'aide de multiplicateurs de Lagrange.
Elle est en tout point semblable à la méthode 'CONTRAINTE' à la différence près qu'elle est de nature totalement itérative et donc très peu gourmande en mémoire. En d'autres termes, le surcoût de stockage lié à la prise en compte du contact est quasiment nul. Cette spécificité a pour effet de la rendre particulièrement adaptée aux cas impliquant des nombres importants de liaisons de contact.

2.3.4.1 Opérande RESI_ABSO

```
♦   RESI_ABSO =  resi,                [R]
```

Ce mot-clé permet de régler la précision de la résolution des inéquations de contact (c'est le critère d'arrêt sur les jeux de l'algorithme itératif). RESI_ABSO (qu'il faut comprendre par « résidu absolu ») représente le niveau d'interpénétration toléré pour les corps en contact. D'un point de vue pratique, il est conseillé lors de la mise au point d'une étude de partir d'une valeur assez grossière (typiquement de l'ordre de la taille des éléments au voisinage des surfaces de contact) et de la diminuer pour obtenir une solution plus précise.

2.3.4.2 Opérande PRE_COND

```
◊   PRE_COND  =  /'SANS',             [DEFAUT]
                 /'DIRICHLET'
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Comme toute méthode de résolution itérative, la méthode 'GCP' peut être accélérée par l'usage d'un préconditionneur. Un seul est aujourd'hui disponible et il s'agit d'un préconditionneur de Dirichlet. Son usage peut dans certains cas accélérer et diminuer sensiblement le temps de résolution.

```
◊   COEF_RESI =   /1.                   [DEFAUT]
                  /coef                 [R]
```

La phase de préconditionnement est réalisée par la résolution itérative d'un problème auxiliaire. Ce mot clé permet de spécifier la précision de la résolution de ce problème par : COEF_RESI*RESI_ABSO .

2.3.4.3 Opérande **RECH_LINEAIRE**

```
◊   RECH_LINEAIRE=   /'ADMISSIBLE'       [DEFAUT]
                     /'NON_ADMISSIBLE'
```

La méthode de résolution 'GCP' nécessite une phase appelée recherche linéaire. Deux variantes sont disponibles : admissible ou pas.

2.3.4.4 Opérande **REAC_ITER**

```
◊    REAC_ITER   =   /3  ,                      [DEFAUT]
                  /reac
```

La méthode de résolution 'GCP' utilise des directions de recherche conjuguées les unes aux autres (dans la même logique qu'un gradient conjugué classique). Toutes les reac itérations, ces directions sont réinitialisées.

2.3.5 Méthode **GLISSIERE**

```
◊   GLISSIERE =   /'NON',                   [DEFAUT]
                  /'OUI',
```

Cette option est disponible uniquement pour la méthode 'CONTRAINTE', la méthode 'CONTINUE' et la méthode 'XFEM'. Elle permet d'activer le mode de contact bilatéral ou en glissière, dans lequel deux surfaces se trouvant en contact restent « collées » (c'est-à-dire avec un jeu nul) quelque soit l'évolution du chargement. Elle autorise de grands glissements relatifs et le mode glissière n'est pas activé avant que les surfaces soient effectivement en contact (elle ne colle pas a priori deux surfaces distantes d'un jeu non nul si le chargement ne l'implique pas).

```
◊   ALARME_JEU    =   /0.0,                 [DEFAUT]
                      /alarm_jeu,           [R]
```

L'opérande 'ALARME_JEU' (disponible uniquement pour la méthode 'CONTRAINTE') permet d'activer une alarme dès que l'algorithme détecte que, sans la méthode glissière, il y aurait décollement des deux surfaces (un jeu virtuel supérieur à zéro). Sa valeur est réglée par défaut à 0, ce qui alarme l'utilisateur dès que les surfaces auraient dû se décoller sans l'option activée.

2.3.6 Choix de la formulation principale pour le frottement

Il n'existe que la formulation du frottement de Coulomb. Pour l'activer, on utilise le mot-clef FROTTEMENT :

```
◊    FROTTEMENT   =   /'SANS'               [DEFAUT]
                      /'COULOMB'
```

Pour donner le coefficient de frottement, on utilise le mot-clef 'COULOMB' :

```
♦    COULOMB   =   /val
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.3.7 Méthode PENALISATION

Cette méthode est une méthode de résolution du contact/frottement par régularisation. Elle est non exacte dans le sens où il y a **toujours** interpénétration lorsque le contact est établi. Si l'on utilise une formulation discrète avec pénalisation, il convient donc de renseigner le ou les coefficients de pénalisation.

◊ E_N= /val_e_n

val_e_n est le coefficient de pénalisation sur l'interpénétration pour la méthode pénalisée. Il est homogène à la raideur de ressorts placés entre les surfaces de contact pour empêcher l'interpénétration. Une valeur de l'ordre du plus grand module d'Young des solides en contact est initialement recommandée. On augmentera la valeur du coefficient jusqu'à l'obtention de résultats stables. En outre il est possible de contrôler les distances d'interpénétration et donc d'affiner son choix de coefficient, ce qui n'est pas le cas du glissement, puisque l'on ne sait pas a priori quelles sont les zones glissantes et non glissantes alors qu'en cas de contact on peut vérifier que les distances d'interpénétration ne sont pas farfelues.

♦ E_T = /val_e_t

val_e_t est le coefficient de pénalisation sur le glissement pour la méthode pénalisée. Il n'est nécessaire que lorsque le frottement est actif. Il faut augmenter la valeur du coefficient jusqu'à l'obtention de résultats stables.

Dans le cas d'une formulation pénalisée en frottement en 3D, on a le paramètre suivant :

```
◊ COEF_MATR_FROT = /0.          [DEFAUT]
                   /val_coef    [R]
```

Ce coefficient, compris entre 0 et 1, permet de modérer l'effet déstabilisant de la partie négative de la matrice de glissement (qui est ajoutée à la rigidité tangente en 3D). Plus ce coefficient est grand, meilleure est la convergence lorsque l'on est proche de l'équilibre et plus la résolution est difficile loin de l'équilibre. Une valeur de 0.5 est donc initialement conseillée. Le défaut de 0 assure la convergence systématique pour un temps de calcul plus long.
Ce coefficient est utilisé pour traiter des contacts surfaciques avec frottement en 3D. Il n'est pas utilisé le reste du temps.

2.3.8 Méthode LAGRANGIEN

C'est une méthode qui diffère de la méthode CONTRAINTE par le fait qu'elle ne dispose pas de preuve de convergence (la mise à jour des contraintes actives se faisant par paquet) et qu'elle est utilisable en FROTTEMENT.
Dans le cas d'une formulation lagrangienne en frottement en 3D, on a le paramètre suivant :

```
◊ COEF_MATR_FROT = /0.          [DEFAUT]
                   /val_coef    [R]
```

Ce coefficient, compris entre 0 et 1, permet de modérer l'effet déstabilisant de la partie négative de la matrice de glissement (qui est ajoutée à la rigidité tangente en 3D). Plus ce coefficient est grand, meilleure est la convergence lorsque l'on est proche de l'équilibre et plus la résolution est difficile loin de l'équilibre. Une valeur de 0.5 est donc initialement conseillée. Le défaut de 0 assure la convergence systématique pour un temps de calcul plus long.
Ce coefficient est utilisé pour traiter des contacts surfaciques avec frottement en 3D. Il n'est pas utilisé le reste du temps.

2.3.9 Méthode CONTINUE

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

La formulation continue est une généralisation des méthodes lagrangienne. Dans la documentation de référence [R5.05.32], on explique les différentes versions. La formulation continue est une formulation lagrangienne pénalisée avec augmentation. Chacun de ces termes (augmentation, régularisation et pénalisation) est donnée par six coefficients (trois pour le contact et trois pour le frottement) :

```
◊  COEF_REGU_CONT  =  /100               [DEFAUT]
                      /coef_regu_cont     [R]
◊  COEF_REGU_FROT  =  /100               [DEFAUT]
                      /coef_regu_frot     [R]
◊  COEF_STAB_CONT  =  /100               [DEFAUT]
                      /coef_stab_cont     [R]
◊  COEF_STAB_FROT  =  /100               [DEFAUT]
                      /coef_stab_frot     [R]
◊  COEF_PENA_CONT  =  /100               [DEFAUT]
                      /coef_pena_cont     [R]
◊  COEF_PENA_FROT  =  /100               [DEFAUT]
                      /coef_pena_frot     [R]
```

Pour choisir les différentes versions, on utilise plutôt deux mots-clefs (un pour le contact et un pour le frottement) :

```
◊  ALGO_CONT   =  /'LAGRANGIEN'     [DEFAUT]
                  /'STABILISE'
                  /'AUGMENTE'
◊  ALGO_FROT   =  /'LAGRANGIEN'     [DEFAUT]
                  /'STABILISE'
                  /'AUGMENTE'
```

L'algorithme `LAGRANGIEN` correspond à la formulation continue « historique » et permet de prendre en compte à la fois l'augmentation des conditions de contact/frottement, leur régularisation et leur stabilisation. Le coefficient est alors unique et il est donné par `COEF_REGU_CONT` (et `COEF_REGU_FROT` en frottement).
L'algorithme `STABILISE` est une version sans terme de pénalisation. L'utilisateur entre donc `COEF_REGU_CONT` et `COEF_STAB_CONT` (ainsi que `COEF_REGU_FROT` et `COEF_STAB_FROT` en frottement).
Enfin, l'algorithme `AUGMENTE` est une version complète comme l'algorithme `LAGRANGIEN` mais avec possibilité d'entrer indépendamment les six coefficients.
Les coefficients de type `REGU` et `STAB` sont relatifs à la régularisation des lois de contact/frottement. Ils peuvent prendre des valeurs de l'ordre de grandeur du pas de temps en dynamique (10^{-5}, 10^{-6}…) jusqu'à beaucoup plus importante (500 par exemple). Ces coefficients agissent sur la vitesse de convergence mais pas sur la précision du résultat.
Les coefficients de type `PENA` permettent d'ajouter un terme de stabilisation qui renforce la coercivité de l'opérateur. Leurs valeurs n'influencent que la convergence. Pour plus de détails, voir [R5.03.52].

2.3.9.1 Opérande `INTEGRATION`

L'opérande `'INTEGRATION'` permet de sélectionner une méthode d'intégration numérique pour les termes de contact et de frottement dans le cas de la formulation continue.

```
◊  INTEGRATION  =  /'NOEUD'            [DEFAUT]
                   /'GAUSS'
                   /'SIMPSON'
                   /'SIMPSON1'
                   /'SIMPSON2'
                   /'NCOTES'
                   /'NCOTES1'
                   /'NCOTES2'
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Plusieurs méthodes sont implémentées ; 'NOEUD' pour un schéma d'intégration aux nœuds, 'GAUSS' pour le schéma classique de Gauss, 'SIMPSON' pour le schéma de Simpson (intégration aux nœuds et aux milieux des éléments) et 'COTES' pour un schéma adaptatif dans le cas du contact linéaire/quadratique.

2.3.9.2 Opérande FORMULATION

L'opérande FORMULATION permet de choisir une formulation du problème en déplacement ou en vitesse dans le cas de la méthode continue. Ce choix concerne l'ensemble du calcul. On l'utilise en dynamique (il n'a aucun sens en statique). En dynamique, l'avantage de la formulation en vitesse est d'éliminer les oscillations numériques de la vitesse et de l'accélération au moment des impacts.
On utilise cette formulation avec un schéma d'ordre 1 en vitesse disponible dans DYNA_NON_LINE dans le mot-clef facteur 'SCHEMA_TEMPS'. On choisira THETA_METHODE avec FORMULATION='VITESSE'. Ce schéma doit être choisi à la place du schéma de Newmark ou HHT. Il nécessite un paramètre THETA qui prend ses valeurs entre 0,5 et 1. THETA=1 donne le maximum d'amortissement numérique.

2.3.9.3 Opérande COMPLIANCE

Cet opérande permet d'activer le modèle de compliance pour la méthode 'CONTINUE'. Ce modèle prend en compte les aspects microscopiques des surfaces (aspérités) et permet une régularisation du modèle de contact de Signorini. En dynamique, l'apport de ce modèle consiste en la possibilité d'introduire une densité de percussion amortissante qui correspond à la dissipation de l'énergie du choc.
La loi de compliance introduite dans *Code_Aster* est une loi polynomiale (voir doc [R5.03.52]).
Les trois paramètres de la loi de compliance sont ASPERITE, E_N et E_V.

```
◊   COMPLIANCE   =  /'NON'            [DEFAUT]
                    /'OUI'
    ♦  ASPERITE  =  /asperite         [R]
     ♦  E_N         =  /e_n              [R]
    ◊  E_V       =  /0.               [DEFAUT]
                    /e_v              [R]
```

2.3.9.4 Opérande USURE

Cet opérande permet, lorsque le frottement est activé, d'utiliser le modèle d'usure d'Archard pour la méthode 'CONTINUE' (voir doc [R5.03.52]). Les deux paramètres de la loi d'Archard sont K et H.

```
◊   USURE     =  /'SANS'              [DEFAUT]
                 /'ARCHARD'
        ♦ K   =  /k                   [R]
        ♦ H   =  /h                   [R]
```

2.3.10 Méthode XFEM

La formulation XFEM est une écriture de type continue. Par rapport à la méthode continue disponible dans *Code_Aster*, on ne dispose que de la version appelée LAGRANGIEN.
On ne modifie donc éventuellement que les coefficients COEF_REGU_CONT et COEF_REGU_FROT, les autres coefficients étant égaux :

```
◊    COEF_REGU_CONT  =  /100                [DEFAUT]
                        /coef_regu_cont     [R]
◊    COEF_REGU_FROT  =  /100                [DEFAUT]
                        /coef_regu_frot     [R]
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.3.10.1 Opérande INTEGRATION

```
◊  INTEGRATION  =  /'FPG4'                [DEFAUT]
                   /'GAUSS'
                   /'FPG2'
                   /'FPG3'
                   /'FPG6'
                   /'FPG7'
                   /'NOEUD'
                   /'SIMPSON'
                   /'SIMPSON1'
                   /'NCOTES'
                   /'NCOTES1'
                   /'NCOTES2'
```

Ce mot-clé a la même signification que celui de la méthode CONTINUE, cf. §2.3.9.1.

2.3.10.2 Opérande ALGO_LAGR

Il existe deux autres paramètres spécifiques à la formulation XFEM. Le premier est le choix de l'algorithme d'élimination des Lagrange de frottement pour satisfaire la condition LBB (voir [R7.02.12]) :

```
◊  ALGO_LAGR =  /'VERSION1'               [DEFAUT]
                /'NON'
                /'VERSION2'
```

Ce paramètre est d'un usage destiné aux experts.

2.3.10.3 Opérande COEF_ECHELLE

La méthode XFEM étant basée sur une formulation hybride à deux champs (déplacements et pressions), dans certaines situations, une grande différence entre les différentes termes conduit à une matrice de rigidité mal conditionnée. En effet, les termes correspondant à la rigidité mécanique sont en $h^3 E$ (avec E le module de Young et h une taille caractéristique de la maille) et les termes correspondant aux relations de contact sont en h^2. Il reste donc un rapport hE entre les deux. Le paramètre 'COEF_ECHELLE' permet d'appliquer un multiplicateur sur les termes correspondants aux Lagrange de contact.

```
◊  COEF_ECHELLE  =  /1.E+6                [DEFAUT]
                    /coef                 [R]
```

La valeur par défaut permet de traiter des problèmes mécaniques avec des produits hE variant entre 1 et 10^{12}. Ce paramètre est d'un usage destiné aux experts.

2.4 Contrôle des algorithmes

Toutes les formulations avec réactualisation géométrique utilisent un algorithme de type point fixe pour résoudre la non-linéarité d'appariement. Les algorithmes résolvant les problèmes en formulation continue ou XFEM utilisent trois boucles de résolution imbriquées pour résoudre la non-linéarité d'appariement, de seuil et de contact.

2.4.1 Opérande REAC_GEOM pour les formulations discrètes

Cet opérande indique sur quelle configuration géométrique est traité le problème de contact.

- REAC_GEOM='AUTO' : on réactualise automatiquement la géométrie *i.e.* le nombre de cycles réactualisation géométrique-itérations jusqu'à convergence n'est pas fixé par avance mais obéit à un critère interne de convergence géométrique. C'est l'option par défaut, conseillée pour résoudre correctement la non-linéarité

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

d'appariement. Elle assure que les conditions de contact ont été imposées sur une configuration (initiale) qui diffère de moins de 5% de la configuration trouvée.

- REAC_GEOM='SANS' : on travaille sur la géométrie initiale. Cette option n'est valable que dans l'hypothèse de petites perturbations ou quand la normale aux surfaces de contact ne change pas au cours du calcul.
- REAC_GEOM='CONTROLE' : lorsque le critère automatique ne parvient pas à être satisfait ou lorsque l'utilisateur souhaite une meilleure précision (c'est-à-dire inférieure à 5%), il doit contrôler lui-même la réactualisation géométrique et pour cela il doit indiquer :
  - NB_REAC_GEOM=n : C'est le nombre de cycles de réactualisations géométriques qui seront effectués par pas de charge. Plaçons-nous à un pas de charge donné.
    - La valeur 1 indique qu'à convergence, on réactualise la géométrie et on passe au pas de charge suivant.
    - La valeur 2 indique qu'à convergence, on ne passe pas au pas de charge suivant. On réactualise la géométrie et on réitère jusqu'à convergence.
    - La valeur n>2 indique que l'on fait n cycles réactualisation géométrique-itérations jusqu'à convergence.

Remarque :

Si vous avez choisi une réactualisation contrôlée avec n>1 et que Code_Aster détecte la nécessité d'une réactualisation géométrique, il vous en avertira par une alarme.
Charge à l'utilisateur de décider si l'erreur commise (nécessairement supérieure à 5%) est acceptable ou non. Il y a en effet un risque d'erreur d'appariement (une maille a été appariée sur une configuration qui a bougé) et donc d'interpénétration.
Dans le cas où l'on résout sur la géométrie initiale ou bien avec un seul cycle de réactualisation géométrique, il n'y a pas d'avertissement car on ne peut pas calculer d'erreur. L'utilisateur doit donc vérifier la validité de son choix.

2.4.2 Opérandes ITER_GEOM_MAXI/ITER_FROT_MAXI/ITER_CONT_MAXI pour méthodes continues

Ces opérandes permettent de contrôler le nombre maximal d'itérations des 3 boucles imbriquées assurant la résolution des non-linéarités liées à la prise en compte de contact-frottement avec une méthode continue.
Rappelons que la boucle géométrique (paramètre ITER_GEOM_MAXI) est une boucle qui sert à réactualiser l'appariement. Le fonctionnement est à peu de choses près identique aux méthodes discrètes. Pour se placer dans un mode automatique, il suffit de fixer un nombre d'itérations maximal élevé, l'algorithme cherche alors à assurer la stationnarité de la géométrie avec un critère de 0,01%. Si ce critère n'est pas atteint au bout du nombre d'itérations fixé, alors la convergence est forcée (ce qui revient à contrôler manuellement le nombre de cycles de réactualisation).
La boucle de frottement (paramètre ITER_FROT_MAXI) est une boucle de point fixe sur le seuil de Coulomb. Le fonctionnement est identique à la boucle de géométrie, le critère est aussi de 0,01%.
Enfin la boucle de contact (paramètre ITER_CONT_MAXI) est une boucle de type contraintes actives qui sert à déterminer les surfaces effectives de contact. On modifie par paquet le statut des nœuds de contact jusqu'à convergence. Si cette convergence n'est pas atteinte au bout du nombre maximal d'itérations, le calcul s'arrête.
Ces opérandes sont utilisables uniquement avec la formulation CONTINUE ou XFEM.

2.4.3 Opérandes NB_RESOL/STOP_SINGULIER/ITER_MULT_MAXI

Ces 3 opérandes sont accessibles pour les méthodes discrètes s'appuyant sur la dualisation des conditions de contact-frottement et la construction d'un complément de Schur (c'est-à-dire CONTRAINTE, LAGRANGIEN et PENALISATION lorsque E_N est non renseigné).

NB_RESOL est le nombre de résolutions simultanées pour la construction du complément de Schur. Réaliser plusieurs résolutions simultanées permet de manipuler des matrices par blocs. Augmenter nb_resol accélère la construction du complément de Schur mais fait perdre de la place mémoire. nb_resol=10 est un bon compromis. Ce paramètre est réservé à un usage **d'expert**.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

STOP_SINGULIER permet de désactiver l'erreur fatale apparaissant si le complément de Schur est singulier suite à une perte importante de décimales dans la factorisation (8 décimales par défaut). On renseigne pour cela STOP_SINGULIER='NON'. Ce paramètre est réservé à un usage **d'expert**.

ITER_MULT_MAXI permet de fixer le nombre maximum d'itérations de l'algorithme de résolution du contact-frottement par activation simultanée de liaisons (c'est-à-dire LAGRANGIEN et PENALISATION lorsque E_N est non renseigné). On ne dispose pas en effet de preuve de convergence dans ce cas. Le nombre d'itérations maximal Nmax est fixé par ITER_MULT_MAXI*Nesclaves où *Nesclaves* est le nombre de nœuds esclaves. Par défaut, ITER_MULT_MAXI est fixé à 4 sauf pour la méthode des contraintes actives où la valeur n'est pas modifiable et reste fixée à 2.
Si on dépasse le nombre maximum d'itérations de contact/frottement, on obtient le message d'erreur 'Echec dans le traitement du contact'. On peut alors tenter de raffiner le maillage, subdiviser le pas de temps, ou changer la valeur de ITER_MULT_MAXI.

2.4.4 Opérandes CONTACT_INIT/SEUIL_INIT

Les opérandes CONTACT_INIT et SEUIL_INIT permettent de fixer respectivement les seuils de démarrage des boucles du contact et du frottement.
En effet, l'hypothèse de base de la résolution par boucles imbriquées est de toujours chercher à résoudre d'abord un problème sans contact (tous les points sont supposés non contactants) et sans frottement (tous les points glissent). Il est néanmoins possible de forcer l'algorithme à prendre en compte le contact initial en renseignant CONTACT_INIT='OUI' et donner un seuil de glissement différent de zéro en renseignant SEUIL_INIT.

Remarque :

le problème de frottement de Coulomb étant résolu par une succession de boucles sur les seuils fixes de Tresca, il est possible de simuler une loi de type Tresca en renseignant SEUIL_INIT. Et en forçant le nombre d'itérations de frottement à 1 (ITER_FROT_MAXI=1).

2.5 Structure de données VALE_CONT

Toutes les méthodes de contact avec ou sans frottement produisent une structure de données de type VALE_CONT avec les composantes suivantes en chaque nœud esclave :

- CONT : indicateur de contact frottant
  - 0 : pas de contact
  - 1 : contact adhérent (uniquement si FROTTEMENT='COULOMB')
  - 2 : contact glissant
- JEU : valeur du jeu
- RN : norme de la réaction normale de contact
- RNX : composante suivant DX de la réaction normale de contact
- RNY : composante suivant DY de la réaction normale de contact
- RNZ : composante suivant DZ de la réaction normale de contact
- GLIX : composante suivant $\mathbf{t}_1$ du glissement tangentiel (repère local)
- GLIY : composante suivant $\mathbf{t}_2$ du glissement tangentiel (repère local)
- GLI : norme du glissement tangentiel
- RTAX : composante suivant DX de la force tangentielle d'adhérence
- RTAY : composante suivant DY de la force tangentielle d'adhérence
- RTAZ : composante suivant DZ de la force tangentielle d'adhérence
- RTGX : composante suivant DX de la force tangentielle de glissement
- RTGY : composante suivant DY de la force tangentielle de glissement
- RTGZ : composante suivant DZ de la force tangentielle de glissement
- RX : composante suivant DX de la force de contact frottant (RNX+RTAX+RTGX)
- RY : composante suivant DY de la force de contact frottant (RNY+RTAY+RTGY)
- RZ : composante suivant DZ de la force de contact frottant (RNZ+RTAZ+RTGZ)

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- R : norme de la force de contact frottant
- I : percussion de la résultante R de la force de contact frottant
- IX : percussion de la composante suivant DX de la force de contact frottant
- IY : percussion de la composante suivant DY de la force de contact frottant
- IZ : percussion de la composante suivant DZ de la force de contact frottant

Elle s'imprime comme suit sous forme de table :

```
MATABLE=POST_RELEVE_T( ACTION=_F(INTITULE='INFOS FROTTEMENT',
                       GROUP_NO='ESCLAVE',
                       RESULTAT=U,
                       INST=10.,
                       TOUT_CMP='OUI',
                       NOM_CHAM='VALE_CONT',
                       OPERATION='EXTRACTION',),);

IMPR_TABLE(TABLE=MATABLE);
```

Il convient de remarquer que les quantités correspondant aux réactions de contact/frottement sont des efforts nodaux (en unité de force) au sens de la formulation EF et non des pressions de contact. En modélisation axisymétrique, tout comme pour l'option FORC_NODA ou REAC_NODA, il faut donc multiplier les valeurs obtenues par 2π pour obtenir la résultante (cf. [U4.81.02]).
Plus généralement, pour avoir accès aux pressions de contact, deux méthodes sont possibles suivant la formulation :

- si l'on est dans le cas d'une formulation discrète, il faut récupérer les contraintes à l'intérieur des éléments finis (SIEF_ELGA) et les extrapoler aux nœuds sur la peau (il y a donc approximation).
- si l'on est dans le cas d'une formulation continue, la pression de contact et la pression de frottement sont accessibles directement comme inconnues du problème (composantes LAGS_C, LAGR_F1 et LAGR_F2 du champ DEPL).

Enfin il faut savoir qu'avec la méthode 'CONTINUE', seul le statut contactant est calculé, c'est-à-dire que CONT vaut 0 (pas de contact) ou 1 (contact adhérent ou glissant).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Opérande LIAISON_UNILATER

3.1 But

Mot clé facteur utilisable pour définir une condition unilatérale (inéquation) de type nodal sur des degrés de liberté quelconques tel que $\sum \alpha_i(t)\,p_i < r(t)$avec p_i la valeur du degré de liberté d'un nœud (déplacement, pression, température), $r(t)$ une valeur numérique imposée (constante ou fonction) et α_i un coefficient réel quelconque.
L'imposition d'une condition unilatérale de type contact mécanique doit se faire par l'opérateur CONTACT. Par rapport à ce dernier, LIAISON_UNILATER est purement nodal et ne procède à aucun appariement des surfaces.
On remarquera que le paramètre α_i permet aussi, si son signe est négatif, d'inverser le sens de l'inégalité.
Ce type de condition n'est utilisable qu'avec les opérateurs non-linéaires STAT_NON_LINE et DYNA_NON_LINE.

3.2 Syntaxe

```
LIAISON_UNILATER    =   _F(
          ◊   METHODE   =  /'CONTRAINTE'              [DEFAUT]

          ♦   /  MAILLE    =  lma1,                   [l_maille]
              /  GROUP_MA  =  lgma1,                  [l_gr_maille]
              /  NOEUD     =  lno1,                   [l_noeud]
              /  GROUP_NO  =  lgno1,                  [l_gr_noeud]

          ◊   SANS_NO       =  lno1,                  [l_noeud]
          ◊   SANS_GROUP_NO=  lno2,                   [l_gr_noeud]

          ♦   NOM_CMP       =  (liste_cmp)            [TXM]

      #   Si AFFE_CHAR_MECA
              ♦  COEF_IMPO    =  (liste_val)          [R]
              ♦  COEF_MULT    =  (liste_val)          [R]
      #   Si AFFE_CHAR_MECA_F
              ♦  COEF_IMPO    =  (liste_fct)          [FONCTION]
              ♦  COEF_MULT    =  (liste_fct)          [FONCTION]

          ◊   NB_RESOL      =  /10                    [DEFAUT]
                               /nbresol               [I]
      )
```

3.3 Opérandes MAILLE/GROUP_MA/NOEUD/GROUP_NO

La condition unilatérale s'exprime sur les nœuds du maillage donnés sous les mot-clefs NOEUD ou GROUP_NO. On peut néanmoins donner les mailles portant les nœuds grâce aux mots-clefs MAILLE et GROUP_MA.
Contrairement au cas du contact; comme les conditions sont nodales, il est inutile d'orienter les normales aux mailles de peau dans le cas LIAISON_UNILATER.

3.4 Opérandes SANS_NOEUD/SANS_GROUP_NO

Ces opérandes permettent d'exclure des nœuds de la liste des nœuds de la même manière que pour l'opérateur CONTACT.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.5 Opérande NOM_CMP

Liste des composantes p_i (degrés de liberté) sur lesquelles s'exerce la relation unilatérale. Ce peut être n'importe quel degré de liberté porté par le nœud. Par exemple : PRE, PRE1, PRE2, TEMP ou encore DX, DY ou DZ.

3.6 Opérande METHODE

L'opérande METHODE permet de sélectionner l'algorithme de résolution du problème sous la condition unilatérale. Pour l'instant seule la méthode des contraintes actives ('CONTRAINTE') est programmée.

3.7 Opérandes COEF_IMPO et COEF_MULT

COEF_IMPO est la valeur $r(t)$ imposée à droite de la relation unilatérale. COEF_MULT est la liste des coefficients multiplicateurs $\alpha(t)$ utilisés devant chacun des ddls. Les longueurs des listes COEF_MULT et NOM_CMP doivent bien entendu être identiques.
Les coefficients $r(t)$ et $\alpha(t)$ peuvent être des réels (AFFE_CHAR_MECA) ou des fonctions (AFFE_CHAR_MECA_F), éventuellement du temps.

3.8 Exemple

On veut imposer la condition 1.3*PRE1-5.2*PRE2 < 4.0, on aura alors :

```
coef_i  = 4.0
coef_m1 = 1.3
coef_m2 =-5.2

NOM_CMP = ('PRE1','PRE2')
COEF_IMPO = coef_i
COEF_MULT = (coef_m1,coef_m2)
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)